

IN THE CLAIMSListing of Claims:

1 Claim 1 (currently amended) A method for stack memory protection comprising the
2 steps of:

3 generating new memory page attributes for a page table used to manage memory,
4 each of said new memory page attributes identifying a block of memory as a new class
5 of memory, each of said new memory page attributes generated by a corresponding new
6 load/store instruction;

7 assigning, by an operating system or a processor, a selected one of said new
8 memory page attributes to a selected block of memory, said selected block of memory
9 used as a new class of memory corresponding to said selected new memory page
10 attribute;

11 blocking normal load /stores to a memory block having one of said new memory
12 page attributes; and

13 blocking a first new load/store to a memory block with one of said new memory
14 page attributes not corresponding to said first new load/store.

1 Claim 2 (currently amended) The method of claim 1, wherein said new classes of
2 memory comprise stack memory with corresponding stack memory load/store
3 instructions.

1 Claim 3 (currently amended) The method in claim 2, wherein a first error condition is
2 generated whenever normal load/stores are attempted to stack memory having a first or
3 a second stack memory attribute with corresponding first or second stack memory
4 load/store instructions.

1 Claim 4 (currently amended) The method in claim 2, wherein a second error condition
2 is generated whenever [[said]] stack memory load/stores are attempted to said memory
3 not having said first or second stack memory attribute.

1 Claim 5 (original) The method in claim 2, wherein a third error condition is generated
2 whenever a stack memory load/store for a first memory stack is attempted to a second
3 memory stack, said third error condition also generated if a stack memory load/store for
4 said second memory stack is attempted to said first memory stack.

1 Claim 6 (original) The method of claim 2, wherein said stack memory load/store
2 instructions are executed on a CPU comprising an IA64 architecture.

1 Claim 7 (currently amended) The method of claim 5, wherein said first memory stack
2 is a processor stack, said processor stack used by a processor to load and store hardware
3 register contents during program execution, said processor stack [[stacks]] is transparent
4 to a programmer or a compiler.

1 Claim 8 (original) The method of claim 7, wherein said processor stack is an IA64
2 register stack.

1 Claim 9 (original) The method of claim 5, wherein said second memory stack is a
2 program stack, said program stack used by a programmer or a compiler in managing
3 program flow.

1 Claim 10 (original) A processor comprising stack memory protection circuitry, said
2 processor using blocks of memory as stack memory, said stack memory protection
3 circuitry comprising:

4 a stack memory attribute circuit, said stack memory attribute circuit operable to
5 generate memory attributes, said memory attributes associated with each memory block
6 designated as a memory stack;

7 a page table attribute storage circuit, said page table attribute circuit operable to
8 store and associate one of said stack memory attributes with a block of memory
9 designated as stack memory;

10 a stack memory allocation circuit, said stack memory allocation circuit operable
11 to identify a block of memory as a stack memory and associate said memory block with

12 one of said stack memory attributes, said stack memory attributes stored in a memory
13 page table; and

14 a stack memory instruction execution circuit, said stack memory instruction
15 execution circuit operable to decode load/store instructions to memory blocks, said stack
16 memory instruction execution circuit granting stack memory load and stores to memory
17 blocks having a required stack memory attribute and not granting stack memory load and
18 stores to memory blocks not having said required stack memory attribute.

1 Claim 11 (original) The processor in claim 10, wherein a first error condition is
2 generated whenever normal load/stores are attempted to stack memory having a first or
3 a second stack memory attribute.

1 Claim 12 (original) The processor in claim 10, wherein a second error condition is
2 generated whenever said stack memory load/stores are attempted to memory not having
3 a stack memory attribute.

1 Claim 13 (original) The processor in claim 10, wherein a third error condition is
2 generated whenever a stack memory load/store for a first memory stack is attempted to
3 a second memory stack, said third error condition also generated if a stack memory
4 load/store for said second memory stack is attempted to said first memory stack.

1 Claim 14 (original) The processor of claim 10, wherein said stack memory load and
2 store instructions are executed on a CPU comprises an IA64 architecture.

1 Claim 15 (original) The processor of claim 13, wherein said first memory stack is a
2 processor stack, said processor stack used by a processor to load and store hardware
3 register contents during program execution, said processor stacks transparent to a
4 programmer or a compiler.

1 Claim 16 (original) The processor of claim 13, wherein said second memory stack is a
2 program stack, said program stack used by a programmer or a compiler in managing
3 program flow.

1 Claim 17 (original) A data processing system, comprising:

2 a central processing unit (CPU);

3 shared random access memory (RAM);

4 read only memory (ROM);

5 an I/O adapter; and

6 a bus system coupling said CPU to said ROM, said RAM said display adapter,

7 wherein said CPU, said CPU comprising stack memory protection circuitry, said stack

8 memory protection circuitry comprising:

9 a stack memory attribute circuit, said stack memory attribute circuit

10 operable to generate memory attribute, said memory attribute associated with each

11 memory block designated as a memory stack;

12 a page table attribute storage circuit, said page table attribute circuit

13 operable to store and associate said stack memory attribute with a block of memory

14 designated as stack memory;

15 a stack memory allocation circuit, said stack memory allocation circuit

16 operable to identify a block of memory as a stack memory and associate said memory

17 block with a stack memory attribute, said stack memory attribute stored in a memory

18 page table; and

19 a stack memory instruction execution circuit, said stack memory

20 instruction execution circuit operable to decode load/store instructions to memory blocks,

21 said stack memory instruction execution circuit granting stack memory load and stores

22 to memory blocks having a stack memory attribute and not granting stack memory load

23 and stores to memory blocks not having said stack memory attribute.

1 Claim 18 (original) The data processing system in claim 17, wherein a first error

2 condition is generated whenever normal load/stores are attempted to stack memory

3 having a first or a second stack memory attribute.

1 Claim 19 (original) The data processing system in claim 17, wherein a second error
2 condition is generated whenever said stack memory load/stores are attempted to memory
3 not having a stack memory attribute.

1 Claim 20 (original) The data processing system in claim 17, wherein a third error
2 condition is generated whenever a stack memory load/store for a first memory stack is
3 attempted to a second memory stack, said third error condition also generated if a stack
4 memory load/store for said second memory stack is attempted to said first memory stack.

1 Claim 21 (original) The data processing system of claim 17, wherein said stack memory
2 load and store instructions are executed on a CPU comprising an IA64 architecture.

1 Claim 22 (original) The data processing system of claim 20, wherein said first memory
2 stack is a processor stack, said processor stack used by a processor to load and store
3 hardware register contents during program execution, said processor stacks transparent
4 to a programmer or a compiler.

1 Claim 23 (original) The data processing system of claim 20, wherein said second
2 memory stack is a program stack, said program stack used by a programmer or a
3 compiler in managing program flow.

1 Claim 24 (currently amended) A computer program product embodied in a machine
2 readable medium, including an operating system and a compiler for a processor system,
3 comprising; a program of instructions for performing the program steps of:

4 generating new memory page attributes for a page table used to manage memory,
5 each of said new memory page attributes identifying a block of memory as a new class
6 of memory, each of said new memory page attributes generated by a corresponding new
7 load/store instruction;

8 assigning, by an operating system or a processor, a selected one of said new
9 memory page attributes to a selected block of memory, said selected block of memory
10 used as a new class of memory corresponding to said selected new memory page
11 attribute;

12 blocking normal load /stores to a memory block having one of said new memory
13 page attributes; and

14 blocking a first new load/store to a memory block with one of said new memory
15 page attributes not corresponding to said first new load/store,

1 Claim 25 (original) The computer program product of claim 24, wherein said new
2 classes of memory comprise stack memory.

1 Claim 26 (original) The computer program product in claim 25, wherein a first error
2 condition is generated whenever normal load/stores are attempted to stack memory
3 having a first or a second stack memory attribute.

1 Claim 27 (original) The computer program product in claim 25, wherein a second error
2 condition is generated whenever said stack memory load/stores are attempted to memory
3 not having said stack memory attribute.

1 Claim 28 (original) The computer program product in claim 25, wherein a third error
2 condition is generated whenever a stack memory load/store for a first memory stack is
3 attempted to a second memory stack, said third error condition also generated if a stack
4 memory load/store for said second memory stack is attempted to said first memory stack.

1 Claim 29 (original) The computer program product of claim 25, wherein said stack
2 memory load/store instructions are executed on a CPU comprising an IA64 architecture.

1 Claim 30 (original) The computer program product of claim 29, wherein said first
2 memory stack is a processor stack, said processor stack used by a processor to load and
3 store hardware register contents during program execution, said processor stacks
4 transparent to a programmer or a compiler.

1 Claim 31 (original) The computer program product of claim 30, wherein said processor
2 stack is an IA64 register stack.

1 Claim 32 (original) The computer program product of claim 28, wherein said second
2 memory stack is a program stack, said program stack used by a programmer or a
3 compiler in managing program flow.

1 Claim 33 (original) A method of managing a memory device comprising the steps of:
2 partitioning said memory device into a plurality of memory spaces on an
3 as-needed basis; and
4 associating a memory attribute with each memory space; said memory attribute
5 determining a use of each of said memory spaces.

1 Claim 34 (original) The method of claim 33, wherein a particular memory attribute has
2 corresponding load/store instruction.

1 Claim 35 (original) The method of claim 34, wherein a load/store instruction associated
2 with a first memory attribute causes an error condition if attempted on a memory space
3 with a second memory attribute.

1 Claim 36 (original) The method of claim 33, wherein each of said memory attributes
2 are stored in a memory page table, said memory page table used to manage said memory
3 device.